EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	669	compress$4 data byte length	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	WITH	ON	2006/12/14 10:13
L2	7	compress$4 data byte length (storage or stor$4)file	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	WITH	ON	2006/12/14 10:17
L3	2	compress$4 data byte length (storage or stor$4)threshold	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	WITH	ON	2006/12/14 10:19
L4	23	compress$4 data length (storage or stor$4)threshold	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	WITH	ON	2006/12/14 10:27

L5	2	compress$4 data length (storage or stor$4)threshold index$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	WITH	ON	2006/12/14 10:27
L6	2	compress$4 data length (storage or stor$4)threshold index$4 frequenc$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	WITH	ON	2006/12/14 10:28
L7	5	compress$4 data length (storage or stor$4)threshold index$4 frequenc$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	SAME	ON	2006/12/14 10:38
L8	909	compress$4 data length and "341".clas.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	WITH	ON	2006/12/14 10:38
L9	733	compress$4 data length and "341".clas.	USPAT	WITH	ON	2006/12/14 10:41

L10	2	factorial reiterative lossless compress$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	AND	ON	2006/12/14 13:35
L11	19	factorial lossless compress$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	AND	ON	2006/12/14 10:42
L12	2	factorial lossless compress$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	WITH	ON	2006/12/14 10:42
L13	2	factorial reiterative compress$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	AND	ON	2006/12/14 13:52

L14	2	factorial lossless compress$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	WITH	ON	2006/12/14 10:42
L15	2	factorial lossless compress$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	SAME	ON	2006/12/14 10:43
L16	1606	(341/63,50,51).CCLS.	USPAT	OR	OFF	2006/12/14 10:44
L17	2329	(382/233,232,245).CCLS.	USPAT	OR	OFF	2006/12/14 10:44
L18	3322	(358/1.16,1.15).CCLS.	USPAT	OR	OFF	2006/12/14 10:44
L19	498	(341/63,50,51).CCLS.	US-PGPUB	OR	OFF	2006/12/14 10:44
L20	114	(341/51).CCLS.	US-PGPUB	OR	OFF	2006/12/14 10:45
L21	0	(341/51andcompress$4databytelength).CCLS.	US-PGPUB	OR	OFF	2006/12/14 10:45

L22	773	341/51 and compress$4 data byte length	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	AND	ON	2006/12/14 10:51
L23	89	341/51 and (compress$4 data byte length)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	WITH	ON	2006/12/14 10:49
L24	1	341/51 and (compress$4 data byte length)	US-PGPUB	WITH	ON	2006/12/14 10:47
L25	2	"20030090397"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	AND	ON	2006/12/14 10:47
L26	11	341/51 and (compress$4 data byte length)	US-PGPUB	SAME	ON	2006/12/14 10:50
L27	1	341/51 and (compress$4 data byte length)frequency	US-PGPUB	SAME	ON	2006/12/14 10:50

L28	232	compress$4 data byte length	EPO; JPO; DERWEN T; IBM_TD B	SAME	ON	2006/12/14 10:52
L29	82	compress$4 data byte length	EPO; JPO; DERWEN T; IBM_TD B	WITH	ON	2006/12/14 10:52
L30	0	compress$4 data byte length lossless	EPO; JPO; DERWEN T; IBM_TD B	WITH	ON.	2006/12/14 10:52
L31	14	compress$4 data length lossless	EPO; JPO; DERWEN T; IBM_TD B	WITH	ON	2006/12/14 10:54
L32	34	compress$4 data length lossless	EPO; JPO; DERWEN T; IBM_TD B	SAME	ON	2006/12/14 10:55
L33	282	compress$4 lossless.ti.	US-PGPU B; USPAT; USOCR; FPRS; EPO; JPO; DERWEN T; IBM_TD B	WITH	ON	2006/12/14 10:56

L34	147	compress$4 data lossless.ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	WITH	ON	2006/12/14 10:58
L35	2	compress$4 data lossless.ti.	JPO	WITH	ON	2006/12/14 11:01
L36	6	compress$4 lossless.ti.	JPO	WITH	ON	2006/12/14 11:08
L37	44	compress$4 lossless.ti.	US-PGPUB	WITH	ON	2006/12/14 11:08
L38	77	compress$4 lossless.ti.	USPAT	WITH	ON	2006/12/14 11:09
L39	0	JP02006174487	JPO	AND	ON	2006/12/14 13:36
L40	0	JP "02006174487"	JPO	AND	ON	2006/12/14 13:36
L41	0	"02006174487"	JPO	AND	ON	2006/12/14 13:36
L42	0	"02006174487"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	AND	ON	2006/12/14 13:36

L43	2	"2006174487"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	AND	ON	2006/12/14 13:36
L44	8	factorial reiterative	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	AND	ON	2006/12/14 13:53

P❁RTAL
USPTO

Nothing Found

Your search for **+lossless +compression +data+factorial** did not return any results.

You may want to try an Advanced Search for additional options.

Please review the Quick Tips below or for more information see the Search Tips.

Quick Tips

- Enter your search terms in lower case with a space between the terms.

    sales offices

    You can also enter a full question or concept in plain language.

    Where are the sales offices?

- Capitalize proper nouns to search for specific people, places, or products.

    John Colter, Netscape Navigator

- Enclose a phrase in double quotes to search for that exact phrase.

    "museum of natural history" "museum of modern art"

- Narrow your searches by using a + if a search term must appear on a page.

    museum +art

- Exclude pages by using a − if a search term must not appear on a page.

    museum -Paris

Combine these techniques to create a specific search query. The better your description of the information you want, the more relevant your results will be.

    museum +"natural history" dinosaur -Chicago

Useful downloads: 📩 Adobe Acrobat   ⓠ QuickTime   Windows Media Player   Real Player

P☉RTAL
USPTO

Nothing Found

Your search for **+lossless +compression +data+reiterative** did not return any results.

You may want to try an Advanced Search for additional options.

Please review the Quick Tips below or for more information see the Search Tips.

Quick Tips

- Enter your search terms in lower case with a space between the terms.

  ```
  sales offices
  ```

  You can also enter a full question or concept in plain language.

  ```
  Where are the sales offices?
  ```

- Capitalize proper nouns to search for specific people, places, or products.

  ```
  John Colter, Netscape Navigator
  ```

- Enclose a phrase in double quotes to search for that exact phrase.

  ```
  "museum of natural history" "museum of modern art"
  ```

- Narrow your searches by using a **+** if a search term must appear on a page.

  ```
  museum +art
  ```

- Exclude pages by using a **-** if a search term must not appear on a page.

  ```
  museum -Paris
  ```

Combine these techniques to create a specific search query. The better your description of the information you want, the more relevant your results will be.

```
museum +"natural history" dinosaur -Chicago
```

Useful downloads: Adobe Acrobat   QuickTime   Windows Media Player   Real Player

P@RTAL
USPTO

Search: ⦿ The ACM Digital Library   ◯ The Guide

+lossless +data+reiterative | **SEARCH**

Nothing Found

Your search for **+lossless +data+reiterative** did not return any results.

You may want to try an Advanced Search for additional options.

Please review the Quick Tips below or for more information see the Search Tips.

Quick Tips

- Enter your search terms in lower case with a space between the terms.

      sales offices

   You can also enter a full question or concept in plain language.

      Where are the sales offices?

- Capitalize proper nouns to search for specific people, places, or products.

      John Colter, Netscape Navigator

- Enclose a phrase in double quotes to search for that exact phrase.

      "museum of natural history" "museum of modern art"

- Narrow your searches by using a **+** if a search term must appear on a page.

      museum +art

- Exclude pages by using a **−** if a search term must not appear on a page.

      museum -Paris

Combine these techniques to create a specific search query. The better your description of the information you want, the more relevant your results will be.
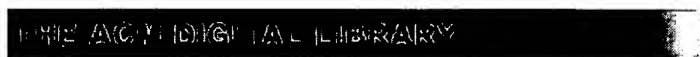
      museum +"natural history" dinosaur -Chicago

Useful downloads: Adobe Acrobat   QuickTime   Windows Media Player   Real Player

PORTAL
USPTO

Search: ⊙ The ACM Digital Library   ○ The Guide

+lossless +compression +data + index   **SEARCH**

THE ACM DIGITAL LIBRARY

ⁱ⁺ Feedback Report a problem Satisfaction survey

Terms used **lossless compression data**   Found **604** of **193,448**

| Sort results by | relevance ▼ | ◆ Save results to a Binder | Try an Advanced Search |
| Display results | expanded form ▼ | ? Search Tips | Try this search in The ACM Guide |

☐ Open results in a new window

Results 1 - 20 of 200   Result page: **1** 2 3 4 5 6 7 8 9 10   next
Best 200 shown   Relevance scale ☐ ⊟ ⊟ ■ ■

1   Energy-aware lossless data compression   ■
◆ Kenneth C. Barr, Krste Asanović
August 2006 **ACM Transactions on Computer Systems (TOCS)**, Volume 24 Issue 3
Publisher: ACM Press
Full text available: 📄 pdf(873.90 KB)   Additional Information: full citation, abstract, references, index terms

> Wireless transmission of a single bit can require over 1000 times more energy than a single computation. It can therefore be beneficial to perform additional computation to reduce the number of bits transmitted. If the energy required to compress data is less than the energy required to send it, there is a net energy savings and an increase in battery life for portable computers. This article presents a study of the energy savings possible by losslessly compressing data prior to transmission. A ...

> **Keywords:** Compression, energy-aware, lossless, low-power, power-aware

2   Lossless compression of volume data   ■
◆ James E. Fowler, Roni Yagel
October 1994 **Proceedings of the 1994 symposium on Volume visualization**
Publisher: ACM Press
Full text available: 📄 pdf(798.03 KB)   Additional Information: full citation, citings, index terms

3   Energy aware lossless data compression   ■
◆ Kenneth Barr, Krste Asanović
May 2003 **Proceedings of the 1st international conference on Mobile systems, applications and services MobiSys '03**
Publisher: ACM Press
Full text available: 📄 pdf(299.94 KB)   Additional Information: full citation, abstract, references

> Wireless transmission of a bit can require over 1000 times more energy than a single 32-bit computation. It would therefore seem desirable to perform significant computation to reduce the number of bits transmitted. If the energy required to compress data is less than the energy required to send it, there is a net energy savings and consequently, a longer battery life for portable computers. This paper reports on the energy of lossless data compressors as measured on a StrongARM SA-110 system. W ...

4 Lossless compression of computer generated animation frames

Hee Cheol Yun, Brian K. Guenter, Russell M. Mersereau
October 1997 **ACM Transactions on Graphics (TOG)**, Volume 16 Issue 4
Publisher: ACM Press

Full text available: 📄 pdf(5.18 MB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>, <u>review</u>

This article presents a new lossless compression algorithm for computer animation image sequences. The algorithm uses transformation information available in the animation script and floating point depth and object number information at each pixel to perform highly accurate motion prediction with vary low computation. The geometric data (i.e., the depth and object number) can either be computed during the original rendering process and stored with the image or computed on the fly during com ...

Keywords: compression, computer animation, computer graphics, motion prediction

5 Segmentation-based multilayer diagnosis lossless medical image compression

Xin Bai, Jesse S. Jin, Dagan Feng
June 2004 **Proceedings of the Pan-Sydney area workshop on Visual information processing VIP '05**
Publisher: Australian Computer Society, Inc.

Full text available: 📄 pdf(380.07 KB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>index terms</u>

Hospital and clinical environments are moving towards computerisation, digitisation and centralisation, resulting in prohibitive amounts of digital medical image data. Compression techniques are, therefore, essential in archival and communication of medical image. Although lossy compression yields much higher compression rates, the medical community has relied on lossless compression for legal and clinical reasons. In this paper, we propose a segmentation-based multilayer (SML) coding scheme for ...

Keywords: lossless compression, medical image coding, region-based image processing, segmentation

6 Performance optimization of wireless local area networks through VLSI data compression

Bongjin Jung, Wayne P. Burleson
January 1998 **Wireless Networks**, Volume 4 Issue 1
Publisher: Kluwer Academic Publishers

Full text available: 📄 pdf(664.69 KB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

In contrast to wireline communication, the physical bandwidth of RF wireless communication systems is relatively limited and is unlikely to grow significantly in the future. Hence it is advantageous to increase the effective bandwidth of communication channels at the expense of complex processing at both the sending and receiving entities. In this paper we present a real-time, low-area, and low-power VLSI lossless data compressor based on the first Lempel-Ziv algorithm (Ziv and Lempel, 1977 ...

7 In-network processing: Data compression algorithms for energy-constrained devices in delay tolerant networks

Christopher M. Sadler, Margaret Martonosi
October 2006 **Proceedings of the 4th international conference on Embedded networked sensor systems SenSys '06**
Publisher: ACM Press

Full text available: pdf(428.25 KB)　Additional Information: full citation, abstract, references, index terms

Sensor networks are fundamentally constrained by the difficulty and energy expense of delivering information from sensors to sink. Our work has focused on garnering additional significant energy improvements by devising computationally-efficient lossless compression algorithms on the source node. These reduce the amount of data that must be passed through the network and to the sink, and thus have energy benefits that are multiplicative with the number of hops the data travels through the networ ...

Keywords: data compression, energy efficient communications, mobile ad hoc networks, wireless sensor networks

8 Constraints in data mining: *SPARTAN*: using constrained models for guaranteed-error semantic compression
Shivnath Babu, Minos Garofalakis, Rajeev Rastogi
June 2002 **ACM SIGKDD Explorations Newsletter**, Volume 4 Issue 1
Publisher: ACM Press
Full text available: pdf(259.12 KB)　Additional Information: full citation, abstract, references, citings

While a variety of lossy compression schemes have been developed for certain forms of digital data (e.g., images, audio, video), the area of lossy compression techniques for arbitrary data tables has been left relatively unexplored. Nevertheless, such techniques are clearly motivated by the ever-increasing data collection rates of modern enterprises and the need for effective, guaranteed-quality approximate answers to queries over massive relational data sets.In this paper, we propose *SPARTAN* ...

9 Bioinformatics (BIO): MACE: lossless compression and analysis of microarray images
Robert Bierman, Nidhi Maniyar, Charles Parsons, Rahul Singh
April 2006 **Proceedings of the 2006 ACM symposium on Applied computing SAC '06**
Publisher: ACM Press
Full text available: pdf(517.72 KB)　Additional Information: full citation, abstract, references, index terms

The ubiquity of microarray expression data in state-of-the-art biology has been well established. The widespread adoption of this technology coupled with the significant volume of image-based experimental data generated per experiment (averaging 40 MB), have led to significant challenges in storage and query-retrieval of primary data from microarray experiments. Research in the yet nascent area of microarray data-compression seeks to address this problem. In this paper, we propose a conceptually ...

Keywords: microarray, microarray data analysis, microarray data compression and storage

10 SPARTAN: a model-based semantic compression system for massive data tables
Shivnath Babu, Minos Garofalakis, Rajeev Rastogi
May 2001 **ACM SIGMOD Record , Proceedings of the 2001 ACM SIGMOD international conference on Management of data SIGMOD '01**, Volume 30 Issue 2
Publisher: ACM Press
Full text available: pdf(240.19 KB)　Additional Information: full citation, abstract, references, citings, index terms

While a variety of lossy compression schemes have been developed for certain forms of digital data (e.g., images, audio, video), the area of lossy compression techniques for arbitrary data tables has been left relatively unexplored. Nevertheless, such techniques are clearly motivated by the ever-increasing data collection rates of modern enterprises

and the need for effective, guaranteed-quality approximate answers to queries over massive relational data sets. In this paper, we propose *SPA ...*

11 Wavelets applied to loseless compression and progressive transmission of floating point data in 3-D curvilinear grids

Aaron Trott, Robert Moorhead, John McGinley

October 1996 **Proceedings of the 7th conference on Visualization '96**

Publisher: IEEE Computer Society Press

Full text available: pdf(493.37 KB)
Publisher Site

Additional Information: full citation, references, citings, index terms

12 Lossless image compression using pixel reordering

Michael Ciavarella, Alistair Moffat

January 2004 **Proceedings of the 27th Australasian conference on Computer science - Volume 26 ACSC '04**

Publisher: Australian Computer Society, Inc.

Full text available: pdf(133.88 KB) Additional Information: full citation, abstract, references, index terms

Lossless image compression techniques typically consider images to be a sequence of pixels in row major order. The processing of each pixel consists of two separate operations. The first step forms a prediction as to the numeric value of the next pixel. Typical predictors involve a linear combination of neighboring pixel values, possibly in conjunction with an edge detection heuristic. In the second step, the difference between that prediction and the actual value of the next pixel is coded. In ...

Keywords: Burrows-Wheeler transformation, image compression, pixel reordering

13 Efficient end to end data exchange using configurable compression

Yair Wiseman, Karsten Schwan, Patrick Widener

July 2005 **ACM SIGOPS Operating Systems Review,** Volume 39 Issue 3

Publisher: ACM Press

Full text available: pdf(10.77 MB) Additional Information: full citation, abstract, references, index terms

We explore the use of compression methods to improve the middleware-based exchange of information in interactive or collaborative distributed applications. In such applications, good compression factors must be accompanied by compression speeds suitable for the data transfer rates sustainable across network links. Our approach combines methods that continuously monitor current network and processor resources and assess compression effectiveness, with techniques that automatically choose suitable ...

Keywords: communication lines, compression

14 Optimal prefetching via data compression

Jeffrey Scott Vitter, P. Krishnan

September 1996 **Journal of the ACM (JACM),** Volume 43 Issue 5

Publisher: ACM Press

Full text available: pdf(564.53 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Caching and prefetching are important mechanisms for speeding up access time to data on secondary storage. Recent work in competitive online algorithms has uncovered several promising new algorithms for caching. In this paper, we apply a form of the

competitive philosophy for the first time to the problem of prefetching to develop an optimal universal prefetcher in terms of fault rate, with particular applications to large-scale databases and hypertext systems. Our prediction algorithms wit ...

Keywords: Markov source, caching, competitive analysis, data compression, databases, fault rate, hypertext, prediction, prefetching, secondary stage, universal prefetcher

15 Bayesian networks for lossless dataset compression
Scott Davies, Andrew Moore
August 1999 **Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining**
Publisher: ACM Press
Full text available: pdf(638.07 KB)   Additional Information: full citation, references, citings, index terms

16 High performance visualization of time-varying volume data over a wide-area network status
Kwan-Liu Ma, David M. Camp
November 2000 **Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM)**
Publisher: IEEE Computer Society
Full text available: pdf(141.35 KB)   Additional Information: full citation, abstract, references, citings, index terms
Publisher Site

This paper presents an end-to-end, low-cost solution for visualizing time-varying volume data rendered on a parallel computer located at a remote site. Pipelining and careful grouping of processors are used to hide I/O time and to maximize processor utilization. Compression is used to significantly cut down the cost of transferring output images from the parallel computer to a display device through a wide-area network. This complete rendering pipeline makes possible highly efficient rende ...

Keywords: Keywords: High Performance Computing, Image Compression,Parallel Volume Rendering, Pipelining, Remote Visualization, Sci-entificVisualization, Time-Varying Data, Wide-Area Network

17 Virtual people & scalable worlds: Efficient compression and delivery of stored motion data for avatar animation in resource constrained devices
Siddhartha Chattopadhyay, Suchendra M. Bhandarkar, Kang Li
November 2005 **Proceedings of the ACM symposium on Virtual reality software and technology VRST '05**
Publisher: ACM Press
Full text available: pdf(537.78 KB)   Additional Information: full citation, abstract, references, index terms

Animation of Virtual Humans (avatars) is done typically using motion data files that are stored on a client or streaming motion data from a server. Several modern applications require avatar animation in mobile networked virtual environments comprising of power constrained clients such as PDAs, Pocket-PCs and notebook PCs operating in battery mode. These applications call for efficient compression of the motion animation data in order to conserve network bandwidth, and save power at the client s ...

Keywords: avatar animation, distributed virtual reality, human motion

18 <u>VLSI architecture for lossless compression of medical images using the discrete wavelet transform</u>
I. Urriza, J. I. Artigas, J. I. García, L. A. Barragán, D. Navarro
February 1998 **Proceedings of the conference on Design, automation and test in Europe**
Publisher: IEEE Computer Society
Full text available: <u>pdf(58.57 KB)</u>
<u>Publisher Site</u>
Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>index terms</u>

This paper presents a VLSI Architecture to implement the forward and inverse 2-D Discrete Wavelet Transform (FDWT/IDWT), to compress medical images for storage and retrieval. Lossless compression is usually required in the medical image field. The word length required for lossless compression makes too expensive the area cost of the architectures that appear in the literature. Thus, there is a clear need for designing an architecture to implement the lossless compression of medical images using ...

Keywords: Medical Image compression, VLSI architectures, DWT

19 <u>Progressive lossless compression of arbitrary simplicial complexes</u>
Pierre-Marie Gandoin, Olivier Devillers
July 2002 **ACM Transactions on Graphics (TOG) , Proceedings of the 29th annual conference on Computer graphics and interactive techniques SIGGRAPH '02**, Volume 21 Issue 3
Publisher: ACM Press
Full text available: <u>pdf(8.88 MB)</u>
Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

Efficient algorithms for compressing geometric data have been widely developed in the recent years, but they are mainly designed for closed polyhedral surfaces which are *manifold* or "nearly manifold". We propose here a *progressive* geometry compression scheme which can handle manifold models as well as "triangle soups" and 3D tetrahedral meshes. The method is lossless when the decompression is complete which is extremely important in some domains such as medical or finite element.Wh ...

Keywords: coding, interactivity, mesh compression, non manifold meshes, progressivity

20 <u>Session P2: large data sets: Interactive rendering of large volume data sets</u>
Stefan Guthe, Michael Wand, Julius Gonser, Wolfgang Straßer
October 2002 **Proceedings of the conference on Visualization '02**
Publisher: IEEE Computer Society
Full text available: <u>pdf(3.21 MB)</u>
Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

We present a new algorithm for rendering very large volume data sets at interactive framerates on standard PC hardware. The algorithm accepts scalar data sampled on a regular grid as input. The input data is converted into a compressed hierarchical wavelet representation in a preprocessing step. During rendering, the wavelet representation is decompressed on-the-fly and rendered using hardware texture mapping. The level of detail used for rendering is adapted to the local frequency spectrum of t ...

Keywords: compression algorithms, level of detail algorithms, scientific visualization, volume rendering, wavelets

Useful downloads: Adobe Acrobat   QuickTime   Windows Media Player   Real Player

Lossless data compression

From Wikipedia, the free encyclopedia

Lossless data compression is a class of data compression algorithms that allows the exact original data to be reconstructed from the compressed data. This can be contrasted to lossy data compression, which does not allow the exact original data to be reconstructed from the compressed data.

Lossless data compression is used in many applications. For example, it is used in the popular ZIP file format and in the Unix tool gzip. It is also often used as a component within lossy data compression technologies.

Lossless compression is used when it is important that the original and the decompressed data be identical, or when no assumption can be made on whether certain deviation is uncritical. Typical examples are executable programs and source code. Some image file formats, notably PNG, use only lossless compression, while others like TIFF and MNG may use either lossless or lossy methods. GIF uses a lossless compression method, but most GIF implementations are incapable of representing full color, so they quantize the image (often with dithering) to 256 or fewer colors before encoding as GIF. Color quantization is a lossy process, but the quantized image can be re-quantized with no further loss. (Some rare GIF implementations make multiple passes over an image, adding 255 new colors on each pass.)

Contents

- 1 Lossless compression techniques
- 2 Lossless compression methods
    - 2.1 Audio compression
    - 2.2 Graphic compression
    - 2.3 Video compression
- 3 Lossless data compression must always make some files longer
- 4 See also
- 5 External links

Lossless compression techniques

Lossless compression methods may be categorized according to the type of data they are designed to compress. The three main types of targets for compression algorithms are text, images, and sound. Whilst, in principle, any general-purpose lossless compression algorithm (general-purpose means that they can handle all binary input) can be used on any type of data, many are unable to achieve significant compression on data that is not of the form that they are designed to deal with. Sound data, for instance, cannot be compressed well with conventional text compression algorithms.

Most lossless compression programs use two different kinds of algorithms: one which generates a *statistical model* for the input data, and another which maps the input data to bit strings using this model in such a way that "probable" (e.g. frequently encountered) data will produce shorter output than "improbable" data. Often, only the former algorithm is named, while the latter is implied (through common use, standardization etc.) or unspecified.

Statistical modeling algorithms for text (or text-like binary data such as executables) include:

- Burrows-Wheeler transform (block sorting preprocessing that makes compression more efficient)
- LZ77 (used by DEFLATE)
- LZW

Encoding algorithms to produce bit sequences are:

- Huffman coding (also used by DEFLATE)
- Arithmetic coding

Many of these methods are implemented in open-source and proprietary tools, particularly LZW and its variants. Some algorithms are patented in the USA and other countries and their legal usage requires licensing by the patent holder. Because of patents on certain kinds of LZW compression, and in particular licensing practices by patent holder Unisys that many developers considered abusive, some open source activists encouraged people to avoid using the Graphics Interchange Format (GIF) for compressing image files in favor of Portable Network Graphics PNG, which combines the LZ77-based deflate algorithm with a selection of domain-specific prediction filters. However, the patents on LZW have now expired. [1] (http://www.unisys.com/about__unisys/lzw).

Many of the lossless compression techniques used for text also work reasonably well for indexed images, but there are other techniques that do not work for typical text that are useful for some images (particularly simple bitmaps), and other techniques that take advantage of the specific characteristics of images (such as the common phenomenon of contiguous 2-D areas of similar tones, and the fact that colour images usually have a preponderance to a limited range of colours out of those representable in the colour space).

As mentioned previously, lossless sound compression is a somewhat specialised area. Lossless sound compression algorithms can take advantage of the repeating patterns shown by the wave-like nature of the data - essentially using models to predict the "next" value and encoding the (hopefully small) difference between the expected value and the actual data. If the difference between the predicted and the actual data (called the "error") tends to be small, then certain difference values (like 0, +1, -1 etc. on sample values) become very frequent, which can be exploited by encoding them in few output bits.

It is sometimes beneficial to compress only the differences between two versions of a file (or, in video compression, of an image). This is called delta compression (from the Greek letter Δ which is commonly used in mathematics to denote a difference), but the term is typically only used if both versions are meaningful outside compression and decompression. For example, while the process of compressing the error in the above-mentioned lossless audio compression scheme could be described as delta compression from the approximated sound wave to the original sound wave, the approximated version of the sound wave is not meaningful in any other context.

Lossless compression methods

For a complete list, see Category:Lossless compression algorithms

Audio compression

- Apple Lossless - ALAC (Apple Lossless Audio Codec)
- Direct Stream Transfer - DST
- Free Lossless Audio Codec - FLAC
- Meridian Lossless Packing - MLP
- Monkey's Audio - Monkey's Audio APE
- RealPlayer - RealAudio Lossless
- Shorten - SHN
- TTA - True Audio Lossless
- WavPack - WavPack lossless
- WMA Lossless - Windows Media Lossless

Graphic compression

- ABO - Adaptive Binary Optimization
- GIF - (lossless, but contains a very limited number color range)
- PNG - Portable Network Graphics
- JPEG-LS - (lossless/near-lossless compression standard)
- JPEG 2000 - (includes lossless compression method, as proven by Sunil Kumar, Prof San Diego State University)
- JBIG2 - (lossless or lossy compression of B&W images)
- TIFF
- WMPhoto - (includes lossless compression method)
- Qbit Lossless Codec - Focuses on intra-frame (single-image) lossless compression

Video compression

- Huffyuv
- SheerVideo
- CorePNG [2] (http://corepng.corecodec.org/)
- MSU Lossless Video Codec
- LCL [3] (http://translate.google.com/translate?u=http%3A%2F%2Fwww.geocities.co.jp%2FPlaytown-Denei%2F2837%2FLRC.htm&langpair=ja%7Cen&hl=en&safe=off&c2coff=1&ie=UTF-8&oe=UTF-8&prev=%2Flanguage_tools)
- Qbit Lossless Codec [4] (http://www.qbit.com/)
- Animation codec
- Lagarith
- H.264/MPEG-4 AVC

Lossless data compression must always make some files *longer*

Lossless data compression algorithms cannot guarantee compression for all input data sets. In other words, for any (lossless) data compression algorithm, there will be an input data set that does not get smaller when processed by the algorithm. This is easily proven with elementary mathematics using a counting argument, as follows:

- Assume that each file is represented as a string of bits of some arbitrary length.
- Suppose that there is a compression algorithm that transforms every file into a distinct file which is no longer than the original file, and that at least one file will be compressed into something that is shorter than itself.
- Let M be the least number such that there is a file F with length M bits that compresses to something shorter. Let N be the length (in bits) of the compressed version of F.
- Because $N < M$, **every** file of length N keeps its size during compression. There are 2^N such files. Together with F, this makes $2^N + 1$ files which all compress into one of the 2^N files of length N.
- But 2^N is smaller than $2^N + 1$, so by the pigeonhole principle there must be some file of length N which is simultaneously the output of the compression function on two different inputs. That file cannot be decompressed reliably (which of the two originals should that yield?), which contradicts the assumption that the algorithm was lossless.
- We must therefore conclude that our original hypothesis (that the compression function makes no file longer) is necessarily untrue.

Any lossless compression algorithm that makes some files shorter must necessarily make some files longer, but it is not necessary that those files become *very much* longer. Most practical compression algorithms provide an "escape" facility that can turn off the normal coding for files that would become longer by being encoded. Then

the only increase in size is a few bits to tell the decoder that the normal coding has been turned off for the entire input. For example, deflate compressed files never need to grow by more than 5 bytes per 65,535 bytes of input.

In fact, if we consider files of length N, if all files were equally probable, then for any lossless compression that reduces the size of some file, the expected length of a compressed file (averaged over all possible files of length N) must necessarily be *greater* than N. So if we know nothing about the properties of the data we are compressing, we might as well not compress it at all. A lossless compression algorithm is only useful when we are more likely to compress certain types of files than others; then the algorithm could be designed to compress those types of data better.

Thus, the main lesson from the argument is not that one risks big losses, but merely that one cannot always win. To choose an algorithm always means implicitly to select a *subset* of all files that will become usefully shorter. This is the theoretical reason why we need to have different compression algorithms for different kinds of files: there cannot be any algorithm that is good for all kinds of data.

The "trick" that allows lossless compression algorithms, used on the type of data they was designed for, to consistently compress such files to a shorter form is that the files the algorithm are designed to act on all have some form of easily-modeled redundancy that the algorithm is designed to remove, and thus belong to the subset of files that that algorithm can make shorter, whereas other files would not get compressed or even get bigger. Algorithms are generally quite specifically tuned to a particular type of file: for example, lossless audio compression programs do not work well on text files, and *vice versa*.

In particular, files of random data cannot be consistently compressed by any conceivable lossless data compression algorithm: indeed, this result is used to *define* the concept of randomness in algorithmic complexity theory.

See also

- Audio data compression
- David A. Huffman
- Information entropy
- Kolmogorov complexity
- Lossless Transform Audio Compression (LTAC)
- Lossy data compression
- List of codecs

External links

- Lossless data compression Benchmarks and Tests (http://www.maximumcompression.com/)
- Comparison of Lossless Audio Compressors (http://wiki.hydrogenaudio.org/index.php?title=Lossless_comparison) at Hydrogenaudio Wiki
- Comparing lossless and lossy audio formats for music archiving (http://www.bobulous.org.uk/misc/audioFormats.html)
- Comparison of Lossless Video Codecs (http://www.compression.ru/video/codec_comparison/lossless_codecs_en.html)
- Links to data compression topics and tutorials (http://datacompression.info/)
- Catalog of lossless compression sites (http://www.compression-links.info/Lossless)

Retrieved from "http://en.wikipedia.org/wiki/Lossless_data_compression"

Categories: Data compression | Lossless compression algorithms

- This page was last modified 02:03, 5 December 2006.
- All text is available under the terms of the GNU Free Documentation License. (See **Copyrights** for details.) Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc.

LZ77 and LZ78

From Wikipedia, the free encyclopedia
(Redirected from LZ77 and LZ78 (algorithms))

LZ77 and **LZ78** are the names for the two lossless data compression algorithms published in papers by Abraham Lempel and Jacob Ziv in 1977 and 1978. These two algorithms form the basis for most of the LZ variations including LZW, LZSS and others. They are both dictionary coders, unlike minimum redundancy coders or run length coders. LZ77 is the "sliding window" compression algorithm, which was later shown to be equivalent to the explicit dictionary technique first given in LZ78.

Contents

- 1 LZ77
- 2 LZ78
- 3 References
- 4 External links

LZ77

LZ77 algorithms achieve compression by replacing portions of the data with references to matching data that has already passed through both encoder and decoder. A match is encoded by a pair of numbers called a *length-distance pair*, which is equivalent to the statement "each of the next *length* characters is equal to the character exactly *distance* characters behind it in the uncompressed stream." (The "distance" is sometimes called the "offset" instead.)

The encoder and decoder must both keep track of some amount of the most recent data, such as the last 2 KB, 4 KB, or 32 KB. The structure this data is held in is called a *sliding window*, which is why LZ77 is sometimes called **sliding window compression**. The encoder needs to keep this data to look for matches, and the decoder needs to keep this data to interpret the matches the encoder refers to. This is why the encoder can use a smaller size sliding window than the decoder, but not vice-versa.

Many documents which talk about LZ77 algorithms describe a length-distance pair as a command to "copy" data from the sliding window: "Go back *distance* characters in the buffer and copy *length* characters, starting from that point." While those used to imperative programming may find this model intuitive, it may also make it hard to understand a feature of LZ77 encoding: namely, that it is not only acceptable but frequently useful to have a length-distance pair where the length actually exceeds the distance. As a copy command, this is puzzling: "Go back *one* character in the buffer and copy *seven* characters, starting from that point." How can seven characters be copied from the buffer when only one of the specified characters is actually *in* the buffer? Looking at a length-distance pair as a statement of identity, however, clarifies the confusion: each of the next seven characters is identical to the character that comes one before it. This means that each character can be determined by looking back in the buffer -- even if the character looked back to was not *in* the buffer when the decoding of the current pair began. Since by definition a pair like this will be repeating a sequence of *distance* characters multiple times, it means that LZ77 incorporates a flexible and easy form of run-length encoding.

Even though all LZ77 algorithms work by definition on the same basic principle, they can vary widely in how they output their encoded data -- what values are possible for lengths and distances, for example, and how length-distance pairs are distinguished from *literals* (single characters encoded as themselves, rather than as part of a length-distance pair.) A few examples:

- The algorithm illustrated in Lempel and Ziv's original 1977 paper output all its data three values at a time: the length and distance of the longest match found in the buffer, and the literal which followed that match. If two successive characters in the input stream could only be encoded as literals, the length would be 0.
- In the PalmDoc format, a length-distance pair is always encoded by a two-byte sequence. Of the 16 bits that make up these two bytes, 11 bits go to encoding the distance, 3 go to encoding the length, and the remaining two are used to make sure the decoder can identify the first byte as the beginning of such a two-byte sequence.
- As of 2004, the most popular LZ77 based compression method is called DEFLATE; it combines LZ77 with Huffman coding. Literals, lengths, and a symbol to indicate the end of the current block of data are all placed together into one alphabet. Distances can be safely placed into a separate alphabet; since a distance only occurs just after a length, it cannot be mistaken for another kind of symbol or vice-versa.

LZ78

While the LZ77 algorithm works on past data, the LZ78 algorithm attempts to work on future data. It does this by forward scanning the input buffer and matching it against a dictionary it maintains. It will scan into the buffer until it cannot find a match in the dictionary. At this point it will output the location of the word in the dictionary, if one is available, the match length and the character that caused a match failure. The resulting word is then added to the dictionary.

Though initially popular, the popularity of LZ78 later dampened, possibly because for the first few decades after it was introduced, parts of LZ78 were patent encumbered in the United States. The most popular form of LZ78 compression was the LZW algorithm, a modification of the LZ78 algorithm made by Terry Welch.

References

- Jacob Ziv and Abraham Lempel; *A Universal Algorithm for Sequential Data Compression* (http://www.cs.duke.edu/courses/spring03/cps296.5/papers/ziv_lempel_1977_universal_algorithm.pdf), IEEE Transactions on Information Theory, 23(3), pp.337-343, May 1977.

External links

- List of LZ77 algorithm (and its derivatives) libraries, papers and sources (http://www.compression-links.info/LZSS)

Retrieved from "http://en.wikipedia.org/wiki/LZ77_and_LZ78"

Category: Lossless compression algorithms

Google

lossless data reiterative compression | Search | Advanced Search
Preferences

Web   Results **1 - 10** of about **75** for **lossless data reiterative compression**. (0.60 seconds)

System, method and algorithm for the optimization of entropy for ...
A method for **lossless compression** of input **data**, said method comprising the ... is
directed to a **lossless compression** system and method using **reiterative**, ...
www.freepatentsonline.com/7111094.html - 59k - Cached - Similar pages

Textual substitution **data compression** with finite length search ...
This invention relates to digital **data compression** systems and, more particularly, to
adaptive and invertible or **lossless** digital **data compression** systems. ...
www.freepatentsonline.com/4906991.html - 126k - Cached - Similar pages
[More results from www.freepatentsonline.com]

Data compression system and method patent invention
[0001] The invention relates to the field of **data compression** and in particular ... of **data**
compression based on factorial **reiterative lossless compression**. ...
www.freshpatents.com/Data-compression-system-and-method-
dt20061102ptan20060244639.php - 24k - Cached - Similar pages

ImageMagick - Image Processing and Display Package
When ImageMagick will have more image pixel **data** in memory than the cache ... Image
Quality with **Reiterative** JPEG **Compression**, Volume 39, Number 4, ...
www.rsmas.miami.edu/~angel/web/ImageMagick/ - 46k - Cached - Similar pages

ImageMagick - Image Processing and Display Package
Apply the **lossless** JPEG patch to Independent JPEG Group's source distribution to ...
Image Quality with **Reiterative** JPEG **Compression**, Volume 39, Number 4, ...
www.hpc.unimelb.edu.au/doc/old_www_Magick/install.html - 45k - Cached - Similar pages

Infrared assessment system - US Patent 5775806
Any of the well-known **lossless data compression**/decompression schemes ... To evaluate
the above model, a **reiterative** extrapolation formula was developed. ...
www.patentstorm.us/patents/5775806-description.html - 40k - Cached - Similar pages

[PDF] Integrated
File Format: PDF/Adobe Acrobat
reiterative,. boring, wasting. time,. and. painstaking. ... **data**. set, use. the different.
parameters, ... Based **Lossless/Lossy Compression** ...
ieeexplore.ieee.org/iel5/10826/34120/01625607.pdf - Similar pages

[PDF] Environmental Modelling Environmental Modelling
File Format: PDF/Adobe Acrobat - View as HTML
flows at urban scales (as **reiterative** ... GRIBzip, for the **lossless compression** of.
meteorological **data** stored in the GRIB1. format (ie GRIB Version 1). ...
www.ercim.org/publication/Ercim_News/enw61/EN61.pdf - Similar pages

[PDF] Robust Adaptive Signal Processors
File Format: PDF/Adobe Acrobat
This is used to identify outliers in multidimensional. **data**. **Reiterative** processing for M –
cancellers will be further analyzed, extended, and exploited. ...
scholar.lib.vt.edu/theses/available/etd-04192003-181602/
unrestricted/Dissertation_Picciolo.pdf - Similar pages

Internal Bibliography
Image Quality with **Reiterative** JPEG **Compression**, Kinoshita and Yamamuro ... reported
to be 3% and 5% for each of the sets, with many connections **lossless**. ...
www.cs.kau.se/cs/prtp/allbib.shtml - 839k - Cached - Similar pages

Result Page: **1** 2 3 4 5 6 7   **Next**

Free! Speed up the web. Download the Google Web Accelerator.

| lossless data reiterative compressio | Search |

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

---

Google Home - Advertising Programs - Business Solutions - About Google

©2006 Google